PetitComputer

プチコン

取扱説明書

2 はじめに

プチコンとは

本製品は内蔵されているBASICでプログラムを作成し、実行することができる愉快なBASIC言語学習ツールです。通信機能を使うことで友達にプログラムを渡したり、友達の作った画像をもらうこともできるので、みんなで協力して作ることもできます。

利用規約

- 本製品により作成されたプログラムや画像等のリソースは、送信命令によって多くの人が見たり実行する可能性があります。他人が不快な気持ちになるようなもの、プライベートな情報、他人の権利（肖像権やプライバシー権、著作権など）を侵害するものなどは送信しないでください。
- 公衆に著しく迷惑をかける不良行為を行ったり、わいせつ、名誉棄損にあたる画像を公表した場合、法令や条例によって罰せされる恐れがあります。
- スマイルブームはお客様が送信した情報および送信したプログラム等によって生じたトラブルについては一切責任を持ちません。

2 はじめに

取扱上の注意点

- 本製品に搭載されているBASICは、これまでに発売されているBASICとは、互換性がありません。古い書籍等からの移植の際には、文法の違いにご注意下さい。
- 内部で扱われている数値は、固定小数型を利用しているため計算による誤差が大きく、正確な精度が必要なプログラムには利用できません。
- 本製品では行番号を使った分岐は利用できません。@で始まるラベルを定義した上で分岐命令を利用します。
- 複雑な計算を行ったり、表示する要素を増やすようなプログラムを書くと、実行速度が低下する可能性があります。
- SAVE・RECVFILE・DELETE等のファイルへの書き込みが行われる命令を繰り返し行うと、読み書きに時間がかかるようになることがあります。
- 予約されている命令は、小文字で入力しても自動的に大文字に変換されます。
- 本製品の条件比較は、==で等しい、!=で異なる、となります。(C言語風)
- 本製品のFOR命令は、先に条件を判断します。FOR I=0 TO -1 のようにSTEP1では成り立たない状態の場合、FOR内部を実行せずにスキップしてNEXT以降の命令を実行します。既存BASICのように必ず1回実行されません。
- 本製品に内蔵されているプログラムテキストを編集する機能は、1行100文字、最大9999行まで登録できる仕様ですが、テキスト用に確保されたメモリーの上限に達した時点で入力できなくなります。約52万文字まで入力可能です。
- 画面への描画を行う命令を使った場合、正しく画面に表示されないことがあります。これは、直前に実行していた他のプログラムで、上下画面のページを切り替える命令が下画面を指定していた等が原因で起こります。このような状況になった場合、実行モードに切り替えて、

  ```
  VISIBLE 1,1,1,1,1,1 (Enter)
  GPAGE 0 (Enter)
  GCLS 0 (Enter)
  BGPAGE 0 (Enter)
  BGCLIP 0,0,31,23 (Enter)
  BGOFS 0,0,0 (Enter)
  BGOFS 1,0,0 (Enter)
  SPPAGE 0 (Enter)
  SPCLR (Enter)
  ```

  を実行した上で描画命令を実行してください。これでも出ない場合は、色がすべて見えない値に変更されているか、キャラがすべて透明になっている可能性があります。COLINIT命令やCHRINIT命令を参考に元の状態に戻してください。
- 割り算を含む計算を行い、必要な数値が整数部分だけの場合は、FLOOR 命令を使って整数部分だけを利用して下さい。座標計算などで誤差が積み重なると微妙なズレが生じてしまいます。
- INKEY$() によるキーボードからの情報取得において、TABキーやBSキー等の一部のキーの情報はシステムの都合上取得することができません。

2 はじめに

操作方法

本製品は開発ツールのため、ゲームとは操作方法が異なります。

十字ボタン	テキスト編集モード時にカーソルを移動します。(Lボタンを押しながら上下でページ単位の移動、左右で行頭と行末への移動)
A,B,Y,X,Rボタン	ユーザーがプログラム実行中に利用可能です。Ｙボタンは、テキスト編集モード時バックスペースボタンとして機能します。
Ｌボタン	キーボードにあるSHIFTボタンと同じ扱いとし、キートップ上に紫色で書かれている文字を入力するときに利用します。プログラム実行中はユーザーが利用できます。
STARTボタン	プログラムが登録されている場合、スタートボタンを押すとユーザープログラムが実行されます。プログラム実行中はユーザーが利用できます。
SELECTボタン	ユーザープログラムの実行中、セレクトボタンはESCキーとして機能します。

3 プチコンの使い方(1/3)

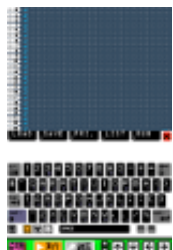
プチコンは以下の３つのモードで構成されています。

●実行モード

キーボードから直接実行可能な命令を入力できます。

▶実行 編集モードから実行ボタンで切替

●編集モード

プログラムのソースコードを書くためのテキスト編集機能です。

編集 実行モードから編集ボタンで切替

●取扱説明書

プチコンの基本的な使い方やBASICの命令一覧を表示します。

説明書ボタンで切替

3 プチコンの使い方(1/3)

モード切替

本製品のモード切替はシステムアイコンで行います。

システムアイコン（実行モード時）

1	取扱説明書アイコン 本製品の使い方を表示します。
2	実行モードアイコン 実行モードへの切り替え。
3	編集モードアイコン テキスト編集モードへの切り替え。
4	ユーザー用システムアイコンエリア プログラム内から利用可能です。

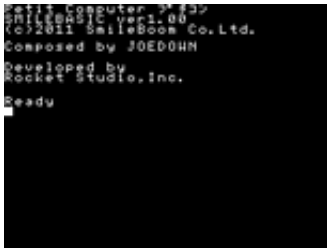
システムアイコン(編集モード時)

5	コピー カーソル位置の1行を内部に取り込みます。
6	ペースト カーソル位置に内部にコピーされている1行を貼り付けます。

3 プチコンの使い方(1/3)

実行モード

「実行モード」は、作ったプログラムを実行するためのモードです。「編集モード」で書いたプログラムは、「実行モード」で、キーボードから"RUN"と入力して、ENTERキーを押すことによって実行することができます。また、作ったプログラムを保存したり、読み込む場合も「実行モード」を利用します。

・上画面：コンソール

キーボードから入力した命令を表示します。

・下画面：キーボード表示

文字切替ボタン	
A	英字
♥	記号
ア	カナ

文字切替ボタンを押すことでキーボード上の文字が変化します。SHIFTキーを押すと、入力できる文字が変化します。
「実行モード」時は、PNLTYPEによるキーボードの切り替えには対応していません。

SAVE命令によって保管されているプログラム等のファイルを消したり、名前を変える場合も「実行モード」から行います。
詳しくは、「12 ◎実行モード専用命令」をご覧ください。

3 プチコンの使い方(1/3)

編集モード

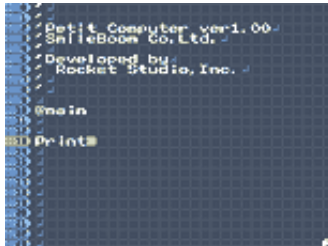
「編集モード」は、プログラムを書くためのモードです。様々な命令を使って自由度の高いプログラムを作ることができます。どのような命令があるかを調べるためには、この説明書を開いて、

「11 命令と関数に関する規則」

：

「31 数学系関数(2/2)」

上記範囲の命令一覧をご利用ください。

・上画面：ソース表示

キーボードから入力した文字列を表示します。改行で自動的に番号が増加します。
1行100文字まで画面に表示し、表示しきれない範囲は横方向へのスクロールで確認します。

・下画面：キーボード表示

「実行モード」と同じように、文字切替ボタンを押すことでキーボード上の文字が変化します。

3 プチコンの使い方(1/3)

プログラムとは

コンピュータは人間からの指示に従って画面への表示や、音を鳴らしたり、タッチスクリーンの状態を調べたりすることができる素晴らしい機械です。

プログラムとは、コンピュータに対して指示を出すための命令の集まりです。本製品に搭載されている「BASIC」と言う名前のコンピュータに指示を出すための言語は、人間が使う言葉に近い文法でコンピュータへ指示を出すことができます。

例えば・・

画面に文字を表示しろ！…PRINT命令
音を鳴らせ！…BEEP命令
画面に円を描け！…GCIRCLE命令
キャラクタを表示しろ！…SPSET命令
数字を記憶しておけ！…変数と代入命令
数値の大小を調べろ！…比較と分岐命令
保存しておけ！…SAVE命令

という感じで、BASICには、コンピュータに指示させたい要素に対応した命令がたくさん用意されています。全ての命令を覚えるのは大変ですが、自分が指示したい命令から少しずつ覚えていけば、ゲームやツールなどを作ることができます。

最初は難しいかもしれませんが、本製品だけあれば、プログラムを作って動かすところまでが実現できます。もちろん間違った命令を書けば、バグと呼ばれる誤動作も生み出せます。

現在、ゲームを作っている会社の人達も小さい頃にBASICを使ってプログラムを覚えた人がたくさんいます。未来のゲーム開発者を目指して、少しずつプログラムの楽しさを味わってみてください。

3 プチコンの使い方(1/3)

簡単なプログラムの書き方

コンソール画面に文字列を表示して音を鳴らすプログラムを書いてみましょう。プチコン起動後、サンプル等を実行していない状態で、以下の手順を実行してください。（最初はプログラムが無い状態で実行モードになっています）

1) 編集ボタンを押して編集モードにする
2) 1行目に PRINT "MOJIDESU" と入力
3) 2行目に BEEP と入力

```
0001 PRINT "MOJIDESU"
0002 BEEP
```

4) 実行ボタンを押して実行モードにする
5) RUN (Enter) と入力

MOJIDESU
OK

コンソールに文字が表示されてピッと音が出ます。また、実行モードで直接命令を実行することも可能です。

GLINE 0,0,255,191,15 （Enter）

これで、上画面に左上から右下へ斜めの線が描画されます。

GCLS (Enter)

と入力すると文字は消えずに線だけを消すことができます。どのような動きをするのか、ちょっと試してみたい時には、実行モードで命令を入力して試してみましょう。

4 プチコンの使い方(2/3)

表示画面構成

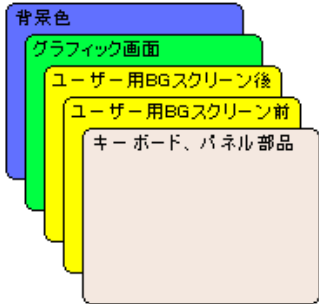
上画面

画面最奥から、背景、グラフィック画面、BGスクリーン後、BGスクリーン前、コンソール画面の5つの画面で構成されています。

ユーザープログラム実行中、SPSETおよびSPCHR命令でグラフィック画面、BG後、BG前の前後にSPRITEの表示優先順位を変更することができます。

背景色	全ての画面の後ろに表示される単色面
グラフィック	線や円を描いたり色を塗ったりすることができる画面
ユーザー用BGスクリーン後	後ろに表示されるBGスクリーン面（8x8キャラを64x64個分管理）
ユーザー用BGスクリーン前	手前に表示されるBGスクリーン面（8x8キャラを64x64個分管理）
コンソール画面	コンソール画面をBG2枚で表現

下画面

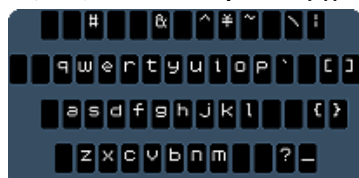
画面最奥から、背景、グラフィック画面、BGスクリーン後、BGスクリーン前、キーボード、パネル部品の5つの画面で構成されています。普段はキーボード表示用画面。プログラム実行時にPNLTYPE命令でBGやグラフィックを利用することができます。

背景色	全ての画面の後ろに表示される単色面
グラフィック画面	線や円を描いたり色を塗ったりすることができる画面
ユーザー用BG 前 後	ユーザー用BGスクリーン
キーボード、パネル部品	キーボード、操作パネル、およびファイルダイアログ等の表示

4 プチコンの使い方(2/3)

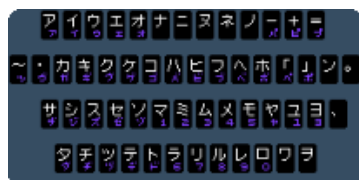
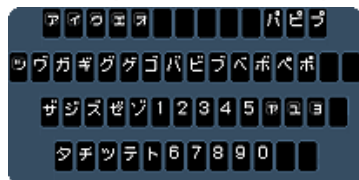
キーボード表

文字列をタッチするとコンソール画面内に対象となる文字列が流し込まれます。SHIFTキー（または本体Ｌボタン）を押すと、入力できる文字列の配置が換わります。

アルファベット

アルファベット+SHIFT（または本体Ｌボタン）

記号

記号+SHIFT（または本体Ｌボタン）

カナ

カナ+SHIFT（または本体Ｌボタン）

4 プチコンの使い方(2/3)

文字コード表

内蔵されている文字は以下の256種類です。
（それぞれ左が16進数、右が10進数のコード）

既存BASICのように0〜31番までのコントロールコードは存在しません。

5 プチコンの使い方(3/3)

カラーパレット

本製品には、コンソールとBGスクリーン、およびSPRITEとグラフィックという表示用要素があります が、

- コンソールの文字とBGスクリーン
- SPRITE
- グラフィック

それぞれに独立したカラーパレットという色の管理構造があります。文字とBGおよびSPRITEの色は、16色（0番は透明）単位で利用することができますが、グラフィックでは256色利用可能です。

BGやSPRITEで使う色は16色ずつに区切って利用し、横方向を色番号、縦をパレット番号と呼びます。

COLSET命令で好きな色に変更することができます。基本的に、左側に濃い色、右に行くほど明るい色を定義しておくことでパレット切り替え時に極端な色の乱れを防ぐことができます。

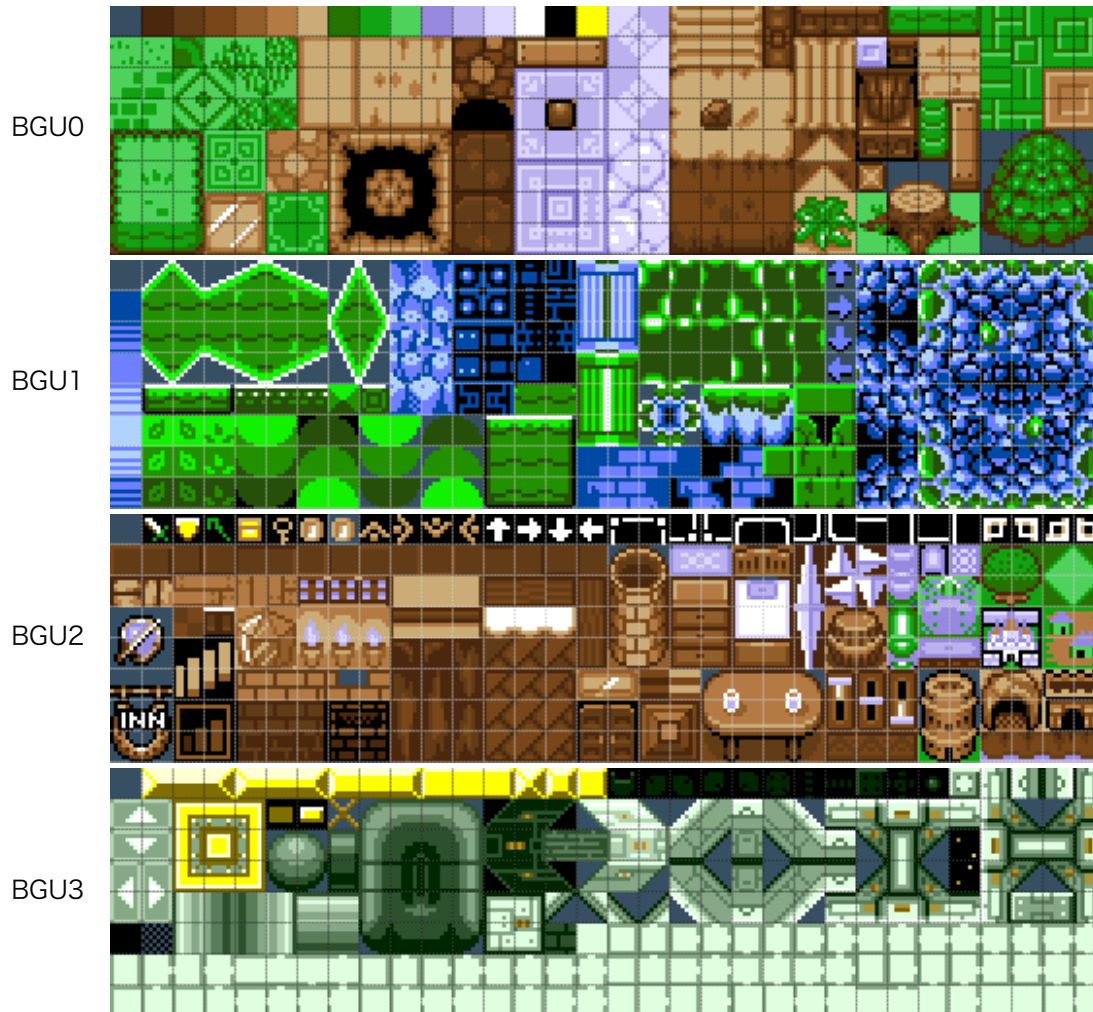
コンソール文字とBG及びSPRITE用

用途ごとに3色ずつ割り当てますが、色の定義は、暗い・普通・明るいになるようにします。

グラフィック用

5 プチコンの使い方(3/3)

ユーザー用BGキャラ

LOADで引数を指定し、画面のBGキャラクタとして利用します。CHRINITで初期状態に戻すことができます。

ユーザー用SPRITEキャラクタ

SPRITE画面上で4方向にキャラを移動するするアニメーションを表示する場合、上下左右の各方向に動くSPRITEキャラ番号をSPUリソースから指定します。
たとえば、SPU1リソース上には、SPRITEキャラ番号64〜67（右）、68〜71（下）、72〜75（左）、76〜79（上）の時計回り順に移動アニメーション画像が並んでいます。

原則として4つのSPRITEキャラ番号で一方向の移動アニメーション表示となりますが、ユーザーが自由に書き換えることもできます。また、CHRINITで初期状態に戻すことができます。

SPU1 64〜67（右）

SPU1 68〜71（下）

SPU1 72〜75（左）

SPU1 76〜79（上）

6 サンプルファイル

本製品には以下のサンプルプログラムが含まれています。詳細についてはプログラムの内容をご覧ください。

サンプルファイルの使い方

1) 実行モードにする
2) EXEC"SAMPLE2" (Enterを押す)
3) 停止する場合はセレクトボタンを押す
4) 編集モードでプログラムを確認

EXECの後の""で囲まれた SAMPLE2 の部分を変更することで他のサンプルも確認することができます。なお、含まれている全てのサンプルプログラムについては、SAVE命令を使って別の名前で保存することでスマイルブームに確認することなく自由に改造することが可能です。

基本サンプル

BASICの基本的な命令を利用した学習用のサンプルプログラムです。編集モードでプログラム内容を確認して、BASICでプログラムを作る参考にしてください。

ファイル名	内容
SAMPLE1	コンソールへの文字表示
SAMPLE2	文字入力を使った簡易計算機
SAMPLE3	キーボードを使った簡易鍵盤
SAMPLE4	数あてゲーム
SAMPLE5	バイオリズム
SAMPLE6	下画面を使ったシーケンサー
SAMPLE7	複数の敵を動かして弾で破壊

高度なサンプル

様々なBASIC命令を駆使した高度なサンプルです。各種ツールサンプルは、ゲームを作るときに必要なデータを作る道具としても利用できます。これらのサンプルも編集モードでプログラムを確認することができます。BASICプログラムが理解できるようになったら、自分だけの機能を追加することも可能です。

ファイル名	内容
CHRED	キャラクタ作成ツール
SCRED	BGスクリーン作成ツール
GRPED	256色画像作成ツール
GAME1	画面上のドットを全て消すレーシングゲーム
GAME2	敵を倒しながらダンジョンを進むロールプレイングゲーム
GAME3	宇宙空間で戦う縦スクロールシューティングゲーム

7 高度なサンプル詳細

基本サンプルも含めて、すべてのサンプルはプログラムを編集することが可能です。ツールに関しては基本的な機能しか搭載されておりませんが、プログラムを改造することで、必要な機能を追加することもできます。

キャラ作成ツール

EXEC "CHRED" Enter で実行。

BGやSPRITE用のキャラデータを作成するツールです。色見本やツールを選択して、EDITエリアに描き込みます。Aボタンを押してファイルモードに入り、L（またはS）を入力してENTERを押すと読み込み（書き込み）を行います。保存したデータはプログラムから利用可能です。

(例) LOAD "BGU0:MYBG00"

BGスクリーン作成ツール

EXEC "SCRED" Enter で実行。

背景用BGキャラを並べて町並み等のBGスクリーンデータを作成するツールです。
キャラ見本からキャラを選んでEDITエリアへ貼り付けます。
保存したデータはプログラムから利用可能です。

(例) LOAD "SCU0:MYSC00"

グラフィック作成ツール

EXEC "GRPED" Enter で実行。

256色まで利用可能なグラフィック画面用の画像データを作成するツールです。
色見本から色を選んでEDITエリアで絵を描きます。
保存したデータはプログラムから利用可能です。

(例) LOAD "GRP0:MYGRP00"

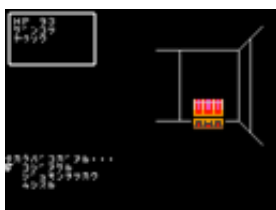
7 高度なサンプル詳細

レーシングゲーム

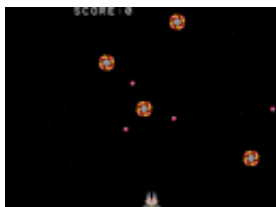
EXEC "GAME1" Enter で実行。

十字ボタンで移動、画面上にあるドットを全て通過するレーシングゲーム。ドットを通過するとスピードアップ。

ロールプレイングゲーム

EXEC "GAME2" Enter で実行。

十字ボタンで移動、迷路を進みモンスターを倒しながら薬草や武器を手に入れて、最上階を目指す。ロールプレイングゲーム。

シューティングゲーム

EXEC "GAME3" Enter で実行。

十字ボタンで移動、Aボタンで発射。
迫りくる敵を倒しながらボスを倒せ！

8 プチコン基本仕様(1/2)

プチコン仕様

プチコンの基本的な仕様一覧です。

基本要素

文字	マルチバイト文字で管理
文字種類	数字、英数字、カナ、記号
数の表現	32ビット固定小数（四捨五入） 4096を1.0として扱う。 整数部は、±524287の範囲。 範囲外の数値は扱えません。
16進表記	&H
2進表記	&B
変数	英字で始まる8文字以内まで、ただし __ は受け付ける。文字列型の変数は、名前の最後に $ をつける。 (例) ANS=75:C$="TEXT"
配列	要素数の合計は32768個、2次元配列まで対応。括弧は () 又は []。添字は0から始まる。 (例) DIM NO(10)の範囲は、 NO(0)〜NO(9)まで。
複数命令	可能（ ： コロンで区切る）
サブルーチンとネスト	制限無し、FOR〜NEXT も同様（メモリーの範囲内）

編集機能

テキスト用のメモリーは約52万文字程度の文字を扱うことができます。これ以上の文字を入力した場合は、最大行数（行内の最大文字数）以内でも登録できなくなります。

エディタ	「編集モード」に搭載される行単位のテキストエディタ。ENTERキーの入力で自動的に行番号が挿入される。テキストの折り返しは行わない。
行の範囲	1〜9999（1行の文字数が多い時、最大行数まで使えない）
行番号の扱い	テキストエディタの行の番号として扱う。改行で自動的に増加し、GOTOやGOSUBなどの命令は、行番号指定はできない仕様となっている。分岐命令については、行番号ではなく「@ラベル名」を利用する。
1行の文字数	1行100文字まで、画面に表示できない範囲は横方向へのスクロールで確認。（テキスト用のメモリーが不足すると1行内の文字数に満たない状態でも入力できなくなります）

8 プチコン基本仕様(1/2)

表示関係

ユーザーSPRITEは、ユーザー用BG面に対して優先順位設定可能。

前 上画面 表示階層 奥	コンソール画面 ユーザー用BG面（前） ユーザー用SPRITE面 ユーザー用BG面（奥） グラフィック面 バックドロップ面
前 下画面 表示階層 奥	キーボードおよびパネル ユーザー用BG面（前） ユーザー用SPRITE面 ユーザー用BG面（奥） グラフィック面 バックドロップ面
グラフィック解像度	256x192ドット
コンソール文字数	32文字x24行（最終行で改行すると1行スクロールする）
アニメ機能	SPRITE命令による簡易アニメ再生

入力装置

キーボード	ソフトウェアキーボード （英字、カナ、記号）
ハードウェアボタン	利用可能（セレクトボタンはESCキーとして利用、Lボタンとスタートボタンは実行時のみ開放）
タッチパネル	システム変数 TCHST、TCHX、TCHY で利用可能

算術演算子

+	加算（A+B）
-	減算（A−B）
*	乗算（A*B）
/	除算（A/B）※0除算はエラー
%	剰余（A%B）※0除算はエラー

8 プチコン基本仕様(1/2)

比較演算子

＞	左辺が右辺より大きい（A＞B）
＜	左辺が右辺より小さい（A＜B）
＞＝	左辺が右辺より大きいか等しい（A＞＝B）※＝＞は禁止
＜＝	左辺が右辺より小さいか等しい（A＜=B）※＝＜は禁止
＝＝	両辺が等しい（A==B）
！＝	両辺が等しくない（A!=B）

ビット演算子

AND	論理積（A AND B）
OR	論理和（A OR B）
XOR	排他的論理和（A XOR B）
NOT	否定（NOT A）

9 プチコン基本仕様(2/2)

演算優先順位

1	（）　[]　で囲まれた部分
2	マイナス　NOT
3	関数
4	＊　／　％（乗除余）
5	＋　－
6	==　!=　<　<=　>　>=
7	AND　OR　XOR

ファイル機能

制御構造	ファイル読み書き中は専用ダイアログが表示されて終わるまで強制待機。
読み書き単位	リソース単位（ユーザーによる自由な読み書きはできません。MEMリソースを利用します）
ファイル名	英字で始まる8文字以内 （'A'～'Z' , '_' , '0'～'9' ）

キャラクタ

リソース（CHR：キャラクタ）のBGスクリーン用とSPRITE用の名称。

ユーザー用BGキャラ	BGU	4バンク(BGU0～BGU3)
ユーザー用SPRITEキャラ	SPU	8バンク(SPU0～SPU7) ※16x16単位で番号管理

カラー

BGとSPRITEは、16色を1パレットと呼び16個のパレットを持つ。

BG用	BG	256色（16色x16種類、ただし種類ごとの0番目は透明）
SPRITE用	SP	256色（16色x16種類、ただし種類ごとの0番目は透明）
グラフィック用	GRP	256色（ただし0番目は透明） 上下画面に共通の256色を割り当てて利用する。

9 プチコン基本仕様(2/2)

サウンド機能

内蔵された音源の再生のみに対応します。

制御要素	プリセット音の再生のみ対応。
同時再生	BGM1曲、効果音4音以上
BGM	0番～29番まで全部で30個の曲が入っています。BGMPLAY 命令で確認して好みの音楽をご利用下さい。 （BGMPLAY命令については、27.◎音楽系命令を参照して下さい） 00　軽快な曲 01　湿った暗い感じの曲 02　緊張感高まる曲 03　激しくアップテンポな曲 04　スタートジングル 05　クリアジングル 06　ゲームオーバー 07　メニューセレクト 08　結果発表 09　スタッフロール 10　スタッフロール　その2 11　時代劇ゲーム風 12　軽快なマーチバンド風 13　激しいロック超 14　軽快な曲　その2 15　WOND 16　考え中 17　WOND2 18　未来系 19　BAL 20　BAL_2 21　スパイ系 22　SCI 23　シューティング 24　パッド 25　SEN 26　ピュア 27　ROA 28　CUR 29　FIG

（表続く）

9 プチコン基本仕様(2/2)

（表続き）

効果音	0番〜69番まで全部で70個の効果音が入っています。BEEP 命令で確認して好みの効果音をご利用下さい。 （BEEP命令については、27.◎音楽系命令を参照して下さい）

番号	効果音	番号	効果音
00	BEEP	35	VOLT
01	ノイズ	36	AUTO
02	カーソル移動	37	キラ
03	決定	38	ESC
04	キャンセル	39	バンジョー2
05	上昇	40	スクラッチ
06	下降	41	ギター2
07	コイン	42	オルガン2
08	ジャンプ	43	ピアノ2
09	着地	44	PASS
10	発射	45	UP2
11	ダメージ	46	録音
12	金属	47	シンセタム
13	爆発	48	カウベル2
14	叫び声	49	metro
15	ブレーキ	50	tri
16	バンジョー	51	コンガ
17	シンセストリングス	52	ダンスBD
18	シンセブラス	53	ダンスSD
19	シンセベース	54	ダンスHH
20	ギター	55	ヒット
21	オルガン	56	ティンバレス
22	ピアノ	57	チャイニーズシンバル
23	カウベル	58	ミニシンバル
24	タム	59	シェーカー
25	シンバル	60	鈴
26	オープンハイハット	61	太鼓
27	クローズハイハット	62	シンセ
28	ハンドクラップ	63	かっこう
29	リムショット	64	パフ！
30	スネアドラム	65	nohkan
31	バスドラム	66	humandr1
32	OK2	67	humandr2
33	BALL	68	犬
34	和風	69	猫

10 システム変数

数値

	R	W	(R=読み込み、W=書き込み)
CSRX	○	×	現在のカーソル横方向位置
CSRY	○	×	現在のカーソル縦方向位置
FREEMEM	○	×	残りユーザー用のメモリー容量
VERSION	○	×	システムのバージョン（0xAABBCCDD、バージョンAA.BB.CC.DD）
ERR	○	×	直前のエラー番号
ERL	○	×	エラー発生行番号
RESULT	○	×	ファイル系命令の実行結果
TCHX	○	×	タッチされたx座標
TCHY	○	×	タッチされたy座標
TCHST	○	×	タッチ状態 (TRUE=触られた)
TCHTIME	○	×	タッチされ続けている時間（フレーム数）
MAINCNTL	○	×	起動時からの経過フレーム時間（最大145分）
MAINCNTH	○	×	起動時からの経過フレーム時間（145分以上の情報）
TABSTEP	○	○	TABによる移動量
TRUE	○	×	必ず1
FALSE	○	×	必ず0
CANCEL	○	×	必ず-1
ICONPUSE	○	○	FALSE=使わない TRUE=使う
ICONPAGE	○	○	ユーザー用システムアイコンのページ番号（実行モードでは常に0が入ります）
ICONPMAX	○	○	ユーザー用システムアイコンのページ最大値(実行モードでは無効)
FUNCNO	○	×	押されているファンクションキーの番号(1～5、0=押されていない)
FREEVAR	○	×	登録可能な変数の数
SYSBEEP	○	○	システム効果音制御（TRUE=あり、FALSE=なし）

文字列

	R	W	(R=読み込み、W=書き込み)
TIME$	○	×	現在の時刻を文字列として取得（HH:MM:SS）
DATE$	○	×	現在の日付を文字列として取得（YYYY/MM/DD）
MEM$	○	○	ファイル保存可能な文字列

11 命令・関数に関する規則

命令方針

あくまでも、コンピュータに対する指示を出すコマンドとして統一した書式を優先します。

◎命令

　命令　[引数] [, 引数] [, 引数] ...

命令の独自書式方針

既存BASICにある命令ごとに変化する書式は引き継がず独自書式を優先させました。

◎既存BASICの視認性を重視したと思われる命令との比較

既存書式

```
LINE( x,y )-( x2,y2 ),color
```

独自書式

```
LINE x,y,x2,y2,color
```

関数方針

独自書式によって命令は'（ 'なしで実現させたため、関数は、命令直後に必ず '（ 'を付けます。

◎関数

```
変数 = COLGET$( "種類名", 番号 )
```

複数の値を取得する命令の方針
（関数と命令の中間的位置づけ）

複数の情報を取得する関数は、C言語であればポインタ渡しが使えますが、BASICでは、ポインタの概念がないため受け取る情報ごとに関数が必要です。本製品では、READの処理系を応用して複数の情報渡しを導入しました。

◎複数値を取得する命令

```
TMREAD( "時間文字列" ) ,H ,M ,S
```

値を取得する場合、引数が必要なケースが多いため引数も渡せるようにしました。受け取る引数の型や、引数の数が変わるような書式には対応しません。

H	時間の値を受け取る変数名
M	分の値を受け取る変数名
S	秒の値を受け取る変数名

◎複数値を取得する命令
　（読み出す引数の指定が複数ある場合）

```
BGREAD( layer, x ,y ) ,N ,C, H, V
```

引数の値によって読み込む情報が変化するため、複雑な読み込みに対応可能です。

12 ◎実行モード専用命令

◎ 実行モード専用命令

プログラムの実行、再開など「実行モード」のみで使用できる5つの命令です（プログラム内記述不可）
NEW、LIST、RUN、CONT、FILES

NEW

プログラムを消す。

書式	NEW	
引数	なし	
戻り	なし	
エラー		

LIST

「編集モード」への切り替えと編集開始。

書式	LIST LIST @ラベル LIST 行番号	
引数	行番号	省略可能、ソースが表示される行の指定
	@ラベル	省略可能、ソースが表示される行の指定
戻り	なし	
エラー		行番号やラベルが存在しない時

RUN

プログラムの実行。

書式	RUN	
引数	なし	
戻り	なし	
エラー		

CONT

STOP命令で停止中のプログラムを再開。

書式	CONT	
引数	なし	
戻り	なし	
エラー		再開できない状態で実行したとき

12 ◎実行モード専用命令

FILES

ファイル一覧をコンソール上に表示。

書式	FILES [リソース名 [, リソース名 ...]]	
引数	リソース名	特定のリソースのみを表示させたい場合に指定
戻り	なし	
エラー		

13 ◎基本命令

◎ 基本命令

メモリーの初期化、配列宣言などを行う以下の命令があります。
CLEAR、＝、DIM、REM、@、KEY、VSYNC

CLEAR

変数名やBASIC内部のメモリーを初期化。

書式	CLEAR	
引数	なし	
戻り	なし	
エラー		

＝ (LET)

代入（ LET 命令自体を省略 ）

書式	ABC=123 TEXT$="ABCDE"	
引数	なし	
戻り	なし	
エラー		

DIM

配列宣言。要素数は2次元まで。
要素数の合計が32768個まで定義可能

書式	DIM pos(4),size(4) DIM sample(10, 5)	
引数	なし	
戻り	なし	
エラー		

' (REM)

注釈、コメント、この記号以降改行までは無視される。

書式	REM コノサキハコメントデス ' comment text	
引数	なし	
戻り	なし	
エラー		

13 ◎基本命令

@（ラベル定義）

必ず行の先頭で利用する。行に対する名前の宣言。GOTOやGOSUBなどで利用。
この記号直後の8文字（スペースは除く）が名前になる、以降改行まで無視。
行番号の代わりに利用する場合は、必ず先頭に@をつけてからLABEL名を書く。

書式	@NAME1 @SAMPLE 区切りは改行か8文字かスペースまで、そのあと改行まで無視。	
引数	ラベル名	文字列とは異なり""で囲む必要はない。
戻り	なし	
エラー		英字、'_'、数字以外の文字や記号を割り当てたとき。ラベル名が指定されていないとき。

KEY

ファンクションキーに文字列を割り当てる。
ユーザープログラム内で利用する場合、押されたファンクションキーに登録された文字列がキー情報として内部的に流し込まれる。

書式	KEY 番号, "文字列"	
引数	番号	ファンクションキーの番号（1〜5）
	文字列	割り当てる文字列（表示上は4文字までだが、内部的には256文字まで確保、表示しきれない場合、最後の文字を '.' として表示）
戻り	なし	
エラー		存在しない番号を指定したとき。

VSYNC

画面更新周期との同期（描画更新待ち）。

書式	VSYNC フレーム数	
引数	フレーム数	直前のVSYNC呼び出しからの経過フレーム数を指定（0=無視）
戻り	なし	
エラー		

14 ◎分岐系命令

◎分岐系命令

分岐条件、ルーチン呼び出しなどを行う以下の命令があります。
ON～GOTO、ON～GOSUB、GOTO、GOSUB、RETURN、STOP、END

ON～　GOTO

数値内容によって分岐。

書式	ON 変数 GOTO 変数=0の時の@ラベル,数値1,数値2...	
引数	なし	
戻り	なし	
エラー		

ON～　GOSUB

数値内容によってサブルーチン呼び出し。

書式	ON 変数 GOSUB 変数=0の時の@ラベル,数値1,数値2...	
引数	なし	
戻り	なし	
エラー		

GOTO

強制分岐。

書式	GOTO @ラベル	
引数	なし	
戻り	なし	
エラー		

GOSUB

サブルーチン呼び出し。

書式	GOSUB @ラベル	
引数	なし	
戻り	なし	
エラー		

14 ◎分岐系命令

RETURN

サブルーチンからの復帰。

書式	RETURN	
引数	なし	
戻り	なし	
エラー		

STOP

実行中のプログラムを強制的に停止してコンソールへ戻る。

書式	STOP	
引数	なし	
戻り	なし	
エラー		

END

プログラムの終了。

書式	END	
引数	なし	
戻り	なし	
エラー		

15 ◎くりかえし・比較命令

◎くりかえし・比較命令

指定回数の繰り返し、条件判断などを行う以下の命令があります。
FOR〜TO〜STEP、NEXT、IF〜THEN、 IF〜GOTO

FOR〜　TO〜　STEP

指定回数の繰り返し（STEPを省略した場合は、STEP1として扱います。増加量がプラスで初期値よりも終了値が小さい時、FOR命令をスキップしてNEXT以降の命令を実行します）

書式	FOR 変数=初期値 TO 終了値 [STEP 増加量]	
引数	なし	
戻り	なし	
エラー		

NEXT

FOR の終わり。

書式	NEXT [変数名]	
引数	なし	
戻り	なし	
エラー		

IF〜　THEN

条件判断（複数行やELSEには対応しない）

書式	IF 条件式 THEN 成立時に実行される命令 IF 条件式 THEN @ラベル	
引数	なし	
戻り	なし	
エラー		

IF〜　GOTO

条件判断（複数行やELSEには対応しない）

書式	IF 条件式 GOTO @ラベル	
引数	なし	
戻り	なし	
エラー		

16 ◎読み込み系命令

◎読み込み系命令

データの読み込みなどを行う以下の命令があります。
READ、DATA、RESTORE、TMREAD()、DTREAD()

READ

DATAの情報を読み込む。

書式	READ 取得変数1 [, 取得変数2 ...] READ A READ B$ READ X,Y,Z,G$	
引数	取得変数...	DATA から読み込む情報を格納する変数。複数指定可能。
戻り	なし	
エラー		読み込むDATAが不足したとき。

DATA

READで読み込むデータの定義（数値も文字も混在可能）

書式	DATA 数値, 数値,... DATA “文字列”, “文字列”,... DATA 123,”SAMPLE”,...	
引数	情報	数値や文字列を ',' で区切って並べる
戻り	なし	
エラー		

RESTORE

READが読み込むDATAの位置を変更する。

書式	RESTORE @ラベル	
引数	なし	
戻り	なし	
エラー		

16 ◎読み込み系命令

TMREAD()

時間文字列を数値に変換。

書式	TMREAD(“時間文字列”), HOUR, MIN, SEC	
引数	時間文字列	HH:MM:SS 形式の時間文字列
	HOUR	時間を受け取る変数
	MIN	分を受け取る変数
	SEC	秒を受け取る変数
戻り	なし	
エラー		

DTREAD()

日付文字列を数値に変換。

書式	DTREAD(“日付文字列”), YEAR, MON, DAY	
引数	日付文字列	YYYY/MM/DD 形式の時間文字列
	YEAR	年を受け取る変数
	MON	月を受け取る変数
	DAY	日を受け取る変数
戻り	なし	
エラー		

17 ◎コンソール基本命令

◎コンソール基本命令

コンソールへの文字表示・変更、文字列・ボタン状態の取得などを行う以下の命令があります。
CLS、COLOR、LOCATE、PRINT、CHKCHR()

CLS

コンソール画面を消去。

書式	CLS	
引数	なし	
戻り	なし	
エラー		

COLOR

コンソールに表示する文字の色指定。

書式	COLOR パレット番号	
引数	パレット番号	0〜15（BG用に割り当てられた16色ずつのパレットの15番の色を利用） 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
戻り	なし	
エラー		

LOCATE

コンソール上の文字表示位置を指定。

書式	LOCATE x座標,y座標	
引数	x座標	横方向の座標（0〜31）※有効範囲外でも受け付ける
	y座標	縦方向の座標（0〜23）※有効範囲外でも受け付ける
戻り	なし	
エラー		

17 ◎コンソール基本命令

PRINT

コンソールへの文字表示。

書式	PRINT “文字列” PRINT 変数 PRINT 変数$ PRINT 変数;変数$;”文字列” PRINT “文字列”,変数,変数 $	
引数	;	複数の要素を続けて表示する場合に利用。
	,	複数の要素を続けて表示するがTAB位置補正がかかる。
戻り	なし	
エラー		

CHKCHR()

コンソール上の文字番号を調べる。

書式	変数 = CHKCHR(x座標,y座標)	
引数	x座標	横方向の座標（0〜31）※有効範囲外でも受け付ける
	y座標	縦方向の座標（0〜23）※有効範囲外でも受け付ける
戻り		-1=範囲外、0〜255＝ASCII文字コード
エラー		

18 ◎コンソール入力系命令

◎コンソール入力系命令

ボタン状態や文字列の取得などを行う以下の命令および関数があります。
BUTTON()、INKEY$()、INPUT、LINPUT

BUTTON()

ハードウェアボタンの状態取得。この命令を使うとビット単位でボタンの同時押し状態を調べることができます。例えば、上と右が同時に押された場合、9が返ります。

書式	変数=BUTTON()	
引数	なし	
戻り	ボタンに対応する番号	
エラー		

番号	ボタン
1	十字ボタンの上
2	十字ボタンの下
4	十字ボタンの左
8	十字ボタンの右
16	Aボタン
32	Bボタン
64	Xボタン
128	Yボタン
256	Lボタン
512	Rボタン
1024	スタートボタン
2048	セレクトボタン

INKEY$()

1文字取得。入力がない場合 "" が戻る

書式	変数$ = INKEY$()	
引数	なし	
戻り	文字変数	キーボードからの1文字が戻る
エラー		

18 ◎コンソール入力系命令

INPUT

数値または文字列の取得。

書式	INPUT “文字列” ; 受け取る変数 INPUT “文字列” ; 受け取る変数$ INPUT “文字列” ; 受け取る変数, 受け取る変数2$	
引数	文字列	省略可能、入力用の説明。
	受け取る変数	キーボードからの入力を受け取るための数値または文字列変数。
	,	複数の変数を指定する場合の区切り。
戻り	なし	
エラー		

LINPUT

文字列の取得。INPUTでは入力できない ',' 等も受け付ける。

書式	LINPUT “文字列” ; 受け取る変数$	
引数	文字列	
	受け取る変数	キーボードから1行分の文字列を受け取る変数。
戻り	なし	
エラー		

19 ◎ファイル系命令

◎ファイル系命令

ファイルの読み込み、保存、消去、を行う以下の命令があります。
LOAD、SAVE、DELETE、EXEC、RENAME

※SAVEやRENAME命令を利用する場合、内蔵されているサンプルで使われているファイル名は利用することができません。

LOAD

ファイルの読み込み(確認ダイアログの表示)

書式	LOAD ”リソース名:ファイル名” [, 表示制御]	
引数	リソース名 読み込む対象となるリソースに割り当てられた文字列	
	PRG	プログラム(省略可能)
	MEM	メモリー
	COL0 〜 COL2	カラー (0=BG, 1=SPRITE, 2=GRP)
	GRP0 〜 GRP1	グラフィック (0=上、1=下)
	SCU0 〜 SCU1	ユーザー用スクリーン (0=手前のレイヤー 1=奥のレイヤー)
	BGU0 〜 BGU3	ユーザー用BGキャラ
	SPU0 〜 SPU7	ユーザー用SPRITEキャラ
	表示制御	FALSE を入れると読み込み時のダイアログ表示を省略
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=キャンセル

19 ◎ファイル系命令

SAVE

ファイルの保存（確認ダイアログの表示）

書式	SAVE "リソース名:ファイル名"	
引数	リソース名	※LOADを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=キャンセル

DELETE

ファイルの消去（確認ダイアログの表示）

書式	DELETE "リソース名:ファイル名"	
引数	リソース名	※LOADを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=キャンセル

EXEC

プログラム内から他のプログラムを読み込んで実行する

書式	EXEC "ファイル名"	
引数	ファイル名	実行するプログラムファイル名
戻り	なし	
エラー	RESULT	FALSE=失敗

RENAME

ファイル名の変更。

書式	RENAME "リソース名:ファイル名", "新しい名前"	
引数	リソース名	※LOADを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=キャンセル

20 ◎通信系命令

◎通信系命令

ファイルの送受信を行う以下の命令があります。
RECVFILE、SENDFILE

RECVFILE

プチコンが保存された他の人のDSi本体からファイルを受信（確認ダイアログの表示）

書式	RECVFILE ”リソース名:ファイル名”	
引数	リソース名	※LOADを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=キャンセル

SENDFILE

プチコンが保存された他の人のDSi本体へファイルを送信（確認ダイアログの表示）

書式	SENDFILE ”リソース名:ファイル名”	
引数	リソース名	※LOADを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=キャンセル

21 ◎描画基本命令

◎描画基本命令

画面表示要素の制御、色指定および情報取得、キャラクタの定義や情報取得、初期化を行う以下の命令があります。

VISIBLE、COLINIT、COLSET、COLREAD()、CHRINIT、CHRSET、CHRREAD()

VISIBLE

画面表示要素の制御（対象面項目を0で非表示、1で表示）

書式	VISIBLE コンソール,パネル,BG0,BG1,SPRITE,グラフィック	
引数	コンソール	0=非表示 1=表示
	パネル	0=非表示 1=表示
	BG0	0=非表示 1=表示
	BG1	0=非表示 1=表示
	SPRITE	0=非表示 1=表示
	グラフィック	0=非表示 1=表示
戻り	なし	
エラー		

COLINIT

色を初期状態に戻す。

書式	COLINIT “色バンク名”, 色番号	
引数	色バンク名 対象を指定する文字列 BG: BGスクリーン用 SP: SPRITE用 GRP: グラフィック用	
	色番号	0〜255
戻り	なし	
エラー		

21 ◎描画基本命令

COLSET

新しい色を指定する（BGの0番は、バックドロップ色となる）

書式	COLSET “色バンク名”, 色番号, “色情報文字列”	
引数	色バンク名	※COLINITを参照
	色番号	※COLINITを参照
	色情報文字列	16進文字列 RRGGBBの順 各要素00〜FFとする （例）"FF00AA”
戻り	なし	
エラー		

COLREAD()

指定した色の情報取得。各要素0〜255

書式	COLREAD(“色バンク名”, 色番号), R, G, B	
引数	色バンク名	※COLINITを参照
	色番号	※COLINITを参照
	R	赤の色要素を受け取る変数
	G	緑の色要素を受け取る変数
	B	青の色要素を受け取る変数
戻り	なし	
エラー		

CHRINIT

指定したキャラクタを初期状態に戻す。

書式	CHRINIT “キャラ名”	
引数	キャラ名 対象キャラクタを指定するための文字列 BGU0〜BGU3　ユーザー用BG用キャラ SPU0〜SPU7　ユーザー用SPRITE用キャラ	
戻り	なし	
エラー		

21 ◎描画基本命令

CHRSET

キャラクタを1つ定義（8x8ドット単位）

書式	CHRSET “キャラ名”, キャラ番号, “画像文字列”	
引数	キャラ名	※CHRINIT参照
	キャラ番号	0〜255
	画像文字列	16色8x8ドット分のキャラクタ情報を16進文字列とする。 （例） “AABBCCDD11223344AABBCCDD11223344AABBCCDD11223344AABBCCDD11223344” （1文字で1ドットを表現）
戻り	なし	
エラー		

CHRREAD()

指定キャラクタの定義情報を取得。

書式	CHRREAD(“キャラ名”, キャラ番号), C$	
引数	キャラ名	※CHRINIT参照
	キャラ番号	※CHRSET参照
	C$	画像文字列を受け取る変数 ※CHRSET参照
戻り	なし	
エラー		

22 ◎SPRITE基本命令

◎SPRITE基本命令

SPRITEの定義、画面消去などを行う以下の命令があります。
SPPAGE、SPSET、SPCLR

SPPAGE

操作対象SPRITE画面の指定。
下画面を選択できますが、下画面は基本的にキーボード用として使われるエリアであり、ユーザー用に用意されたキャラクタを表示させることはできません。あらかじめ用意された簡易の画像表示に利用します。

書式	SPPAGE 画面	
引数	画面	0=上画面 1=下画面
戻り	なし	
エラー		

22 ◎SPRITE基本命令

SPSET

SPRITEの定義(活動開始)
指定管理番号のスプライトが活動を開始します。座標は0,0へ初期化されます。SPSETによって活動を開始した後にキャラ番号だけを変更したい場合は、SPCHR命令を利用します。

書式	SPSET 管理番号,キャラ番号, パレット番号, h反転, v反転, 優先順位	
引数	管理番号	0〜99
	SPRITEキャラ番号	0〜511（下画面への表示の場合0〜117）
	パレット番号	0〜15
	h反転 0＝なし　1=横反転	
	v反転 0＝なし　1=縦反転	
	h反転1,v反転1	
	優先順位 BGに対する表示優先順位を番号で指定。(SPRITE間の表示優先順位は、管理番号が小さい物が手前に表示されます) 0 コンソールの前 1 手前のBGの前 2 2枚のBGの間 3 奥のBGの後	
戻り	なし	
エラー		

SPCLR

SPRITEの消去（活動禁止にする）

書式	SPCLR [管理番号]	
引数	管理番号	0〜99 (省略時全てのSPRITEを消去)
戻り	なし	
エラー		

23 ◎SPRITE制御命令

◎SPRITE制御命令

SPRITEの座標変更やアニメ表示の指定などを行う以下の命令があります。
SPOFS、SPCHR、SPANIM、SPANGLE、SPSCALE、SPCHK()

SPOFS

SPRITE座標の変更。

書式	SPOFS 管理番号, x座標, y座標 [,補間時間]	
引数	管理番号	0〜99
	x座標	-1024〜 +1024 （有効範囲外も受け付ける）
	y座標	-1024〜 +1024 （有効範囲外も受け付ける）
	補間時間	現在の状態と新しい値の間を自動補間する時間（1=1/60秒）
戻り	なし	
エラー		

SPCHR

SPRITEのキャラ番号の変更

書式	SPCHR 管理番号,キャラ番号 [, パレット番号, h反転, v反転, 優先順位]	
引数	管理番号	0〜99
	SPRITEキャラ番号	0〜511（下画面への表示の場合0〜117）
	パレット番号	0〜15
	h反転	0=なし 1=反転
	v反転	0=なし 1=反転
	優先順位	0〜3
戻り	なし	
エラー		

23 ◎SPRITE制御命令

SPANIM

SPRITEによるアニメ表示。
現在指定されているキャラ番号から、この命令で指定された枚数の範囲で指定時間ごとにキャラ番号を切り替える。

書式	SPANIM 管理番号, 枚数, 時間 [, ループ]	
引数	管理番号	0〜99
	枚数	1〜
	時間	1コマを表示する時間（1=1/60秒）
	ループ	0=無限ループ, 1〜（ループ回数）
戻り	なし	
エラー		

SPANGLE

SPRITE角度の変更。回転は左上を原点として実行されるため、回転後の表示位置は角度によって大きく変化します。

書式	SPANGLE 管理番号, 角度 [, 補間時間, 変化方向]	
引数	管理番号	0〜31
	角度	0〜360（範囲外も受け付け）
	補間時間	現在の状態と新しい値の間を自動補間する時間（1=1/60秒）
	変化方向	1=時計回り -1=反時計回り （省略時時計回り）
戻り	なし	
エラー		

23 ◎SPRITE制御命令

SPSCALE

SPRITEスケールの変更。

書式	SPSCALE 管理番号, スケール [, 補間時間]	
引数	管理番号	0〜31
	スケール	0〜200（パーセント相当）
	補間時間	現在の状態と新しい値の間を自動補間する時間（1=1/60秒）
戻り	なし	
エラー		

SPCHK()

自動補間の状態取得。

書式	変数 = SPCHK(管理番号)	
引数	管理番号	0〜99
戻り	状態	FALSE=補間終了 TRUE=補間中
エラー		

24 ◎BGスクリーン命令

◎BGスクリーン命令

操作対象BG画面の指定、表示オフセット変更、指定位置のBGスクリーンへの書き込みなどを行う以下の命令があります。
BGPAGE、BGCLIP、BGOFS、BGPUT、BGREAD()

BGPAGE

操作対象BG画面の指定。

書式	BGPAGE 画面	
引数	画面	0=上画面 1=下画面
戻り	なし	
エラー		

BGCLIP

表示範囲の指定（全てのレイヤーが対象となる）

書式	BGCLIP 始点x, 始点y, 横幅, 縦幅	
引数	始点x座標	0〜31
	始点y座標	0〜23
	終点x座標	0〜31
	終点y座標	0〜23
戻り	なし	
エラー		

BGOFS

BGスクリーンの表示オフセットを変更。

書式	BGOFS レイヤー, x座標, y座標 [, 補間時間]	
引数	レイヤー	0=手前 1=奥
	x座標	横方向の座標（0〜511） （有効範囲外も受け付ける）
	y座標	縦方向の座標（0〜511） （有効範囲外も受け付ける）
	補間時間	現在の状態と新しい値の間を自動補間する時間（1=1/60秒）
戻り	なし	
エラー		

24 ◎BGスクリーン命令

BGPUT

指定位置のBGスクリーンに書き込み。

書式	BGPUT レイヤー, x座標, y座標, キャラ番号, パレット番号, h反転, v反転	
引数	レイヤー	0=手前 1=奥
	x座標	横方向の座標（0〜63） （有効範囲外も受け付ける）
	y座標	縦方向の座標（0〜63） （有効範囲外も受け付ける）
	キャラ番号	0〜1023
	パレット番号	0〜15
	h反転	0＝なし 1=横反転
	v反転	0＝なし 1=縦反転
戻り	なし	
エラー		

BGREAD()

指定位置のBGスクリーンから情報を取得。

書式	BGREAD(レイヤー, x座標, y座標), CHR, PAL, H, V	
引数	レイヤー	0=手前 1=奥
	x座標	横方向の座標（0〜63） （有効範囲外も受け付ける）
	y座標	縦方向の座標（0〜63） （有効範囲外も受け付ける）
	CHR	キャラ番号を受け取る変数
	PAL	パレット番号を受け取る変数
	H	H反転を受け取る変数
	V	V反転を受け取る変数
戻り	なし	
エラー		

25 ◎グラフィック基本命令

◎グラフィック基本命令

操作対象グラフィック画面の指定、消去などを行う以下の命令があります。
GPAGE、GCOLOR、GCLS、GSPOIT()

GPAGE

操作対象グラフィック画面の指定。

書式	GPAGE 画面	
引数	画面	0=上画面 1=下画面
戻り	なし	
エラー		

GCOLOR

グラフィック画面の描画色指定。

書式	GCOLOR 色番号	
引数	色番号	0〜255
戻り	なし	
エラー		

GCLS

操作対象グラフィック画面の消去。

書式	GCLS [色]	
引数	色	0〜255（省略時はGCOLORの色）
戻り	なし	
エラー		

GSPOIT()

指定位置の色を調べる。

書式	変数 = GSPOIT(x座標, y座標)	
引数	x座標	0〜255 （有効範囲外も受け付ける）
	y座標	0〜191 （有効範囲外も受け付ける）
戻り	色	0〜255（範囲外の場合-1）
エラー		

26 ◎グラフィック描画命令

◎グラフィック描画命令

線または円を描く、塗りつぶしなどを行う以下の命令があります。
GPSET、GPAINT、GLINE、GBOX、GFILL、GCIRCLE、GPUTCHR

GPSET

点を打つ。

書式	GPSET x座標, y座標 [,色]	
引数	x座標	0～255 （有効範囲外も受け付ける）
	y座標	0～191 （有効範囲外も受け付ける）
	色	0～255（省略時はGCOLORの色）
戻り	なし	
エラー		

GPAINT

指定位置から塗りつぶし。
(指定位置の色に隣接する同一色が対象)

書式	GPAINT x座標, y座標 [,色]	
引数	x座標	0～255 （有効範囲外も受け付ける）
	y座標	0～191 （有効範囲外も受け付ける）
	色	0～255（省略時はGCOLORの色）
戻り	なし	
エラー		

26 ◎グラフィック描画命令

GLINE

線を引く。

書式	GLINE 始点x座標, 始点y座標, 終点x座標, 終点y座標 [,色]	
引数	始点x座標	0〜255 （有効範囲外も受け付ける）
	始点y座標	0〜191 （有効範囲外も受け付ける）
	終点x座標	0〜255 （有効範囲外も受け付ける）
	終点y座標	0〜191 （有効範囲外も受け付ける）
	色	0〜255（省略時はGCOLORの色）
戻り	なし	
エラー		

GBOX

箱を描く。

書式	GBOX 始点x座標, 始点y座標, 終点x座標, 終点y座標 [,色]	
引数	始点x座標	0〜255 （有効範囲外も受け付ける）
	始点y座標	0〜191 （有効範囲外も受け付ける）
	終点x座標	0〜255 （有効範囲外も受け付ける）
	終点y座標	0〜191 （有効範囲外も受け付ける）
	色	0〜255（省略時はGCOLORの色）
戻り	なし	
エラー		

GFILL

矩形で塗りつぶす。

書式	GFILL 始点x座標, 始点y座標, 終点x座標, 終点y座標 [,色]	
引数	始点x座標	0〜255 （有効範囲外も受け付ける）
	始点y座標	0〜191 （有効範囲外も受け付ける）
	終点x座標	0〜255 （有効範囲外も受け付ける）
	終点y座標	0〜191 （有効範囲外も受け付ける）
	色	0〜255（省略時はGCOLORの色）
戻り	なし	
エラー		

GCIRCLE

円を描く。

書式	GCIRCLE x座標, y座標, 半径 [,色] [, 開始角, 終了角]	
引数	x座標	0〜255 （有効範囲外も受け付ける）
	y座標	0〜191 （有効範囲外も受け付ける）
	半径	0〜255 （有効範囲外も受け付ける）
	色	0〜255（省略時はGCOLORの色）
	開始角	0〜360（有効範囲外も受け付ける）
	終了角	0〜360（有効範囲外も受け付ける）
戻り	なし	
エラー		

GPUTCHR

指定されたキャラクタの画像情報をグラフィック画面に表示。
この命令を利用すると指定したパレット番号の情報がグラフィック用のパレットにコピーされます。定義される場所は、パレット番号 x 16番目の色からの16色分です。

書式	GPUTCHR x座標, y座標, ”キャラ名”, 番号, パレット番号, スケール	
引数	x座標	0〜255 （有効範囲外も受け付ける）
	y座標	0〜191 （有効範囲外も受け付ける）
	キャラ名	※CHRINIT参照
	番号	キャラクタ番号（0〜255）
	パレット番号	キャラクタ用の色（0〜15）
	スケール	拡大率 （1,2,4,8）
戻り	なし	
エラー		

27 ◎音楽系命令

◎音楽系命令

警告音の発声、音楽の演奏開始・停止、演奏状態調査などを行う以下の命令があります。
BEEP、BGMPLAY、BGMSTOP、BGMCHK()

BEEP

単純な警告音（効果音）の発声。

書式	BEEP [波形番号 [,ピッチ [,音量 [,パンポット]]]]	
引数	波形番号	0〜69（省略時は0番）
	ピッチ	-8192で2オクターブ下、0で原音、8192で2オクターブ上 P=4096/12の時の音程変化 C　=P*0 C＃=P*1 D　=P*2 D＃=P*3 E　=P*4 F　=P*5 F＃=P*6 G　=P*7 G＃=P*8 A　=P*9 A＃=P*10 B　=P*11
	音量	0=無音 127=最大
	パンポット	0=左から 64=中央から 127=右から
戻り	なし	
エラー		

BGMPLAY

音楽演奏開始。

書式	BGMPLAY 曲番号	
引数	曲番号	0〜29
戻り	なし	
エラー		

27 ◎音楽系命令

BGMSTOP

音楽演奏停止。

書式	BGMSTOP	
引数	なし	
戻り	なし	
エラー		

BGMCHK()

音楽の演奏状態調査。

書式	変数=BGMCHK()	
引数	なし	
戻り		FALSE=停止中 TRUE=演奏中
エラー		

28 ◎パネル系命令

◎パネル系命令

パネルの種類変更やユーザー用システムアイコンの表示設定、状態確認などを行う以下の命令があります。
PNLTYPE、PNLSTR、
ICONSET、ICONCLR、ICONCHK()

PNLTYPE

パネルの種類変更。

書式	PNLTYPE "パネル名"
引数	パネル名 下画面に表示する種類を決める文字列 OFF パネルがない状態 PNL キーボードがない状態 KYA 英語のキーボード KYM 記号のキーボード KYK カナのキーボード
戻り	なし
エラー	

PNLSTR

下画面への文字列表示（最終行に表示しても改行は発生しません）

書式	PNLSTR x座標,y座標, "文字列" [,パレット番号]	
引数	x座標	横方向の座標（0～31）※有効範囲外でも受け付ける
	y座標	縦方向の座標（0～23）※有効範囲外でも受け付ける
	文字列	表示させたい文字列
	パレット番号	0～15
戻り	なし	
エラー		

ICONSET

ユーザー用システムアイコンの表示キャラクタ設定（及び表示開始）

書式	ICONSET アイコン位置, アイコン番号	
引数	アイコン位置	ユーザー用のシステムアイコン番号（0～3）
	アイコン番号	アイコン専用のキャラ管理番号（0～63）
戻り	なし	
エラー		

28 ◎パネル系命令

ICONCLR

ユーザー用システムアイコンの表示解除。

書式	ICONCLR アイコン位置	
引数	アイコン位置	ユーザー用のシステムアイコン番号（0〜3）
戻り	なし	
エラー		

ICONCHK()

ユーザ用システムアイコンの状態確認。

書式	数値 = ICONCHK()	
引数	なし	
戻り		-1=押されていない 0〜押されたアイコンの番号
エラー		

29 ◎文字系関数

◎文字系関数

文字列から情報を得たり、指定した文字からコードを返すなど、文字に関する以下の関数があります。
ASC()、CHR$()、VAL()、STR$()、HEX$()、MID$()、LEN()

ASC()

指定された文字のASCIIコードを返す。

書式	変数 = ASC(文字)	
引数	なし	
戻り	数値	指定された文字の文字コード
エラー		

CHR$()

指定された文字コードから文字を返す。

書式	変数＄ = CHR$(文字コード)	
引数	文字コード	文字ごとに対応する番号
戻り	文字	文字コードに対応する文字
エラー		

VAL()

文字列から数値を得る。

書式	変数 = VAL(文字列)	
引数	文字列	数値の入った文字列
戻り	数値	文字列から取出した数値
エラー		

STR$()

数値から文字列を得る。

書式	変数$ = STR$(数値)	
引数	数値	文字列にしたい数値
戻り	文字	数値から生成した文字列
エラー		

29 ◎文字系関数

HEX$()

数値から16進文字列を得る。

書式	変数$ = HEX$(数値)	
引数	数値	16進文字列にしたい数値
戻り	文字	数値から生成した16進文字列
エラー		

MID$()

文字列の開始位置から指定数分の文字列を取り出す

書式	変数$ = MID$(文字列, 開始位置, 文字数)	
引数	文字列	元になる文字列
	開始位置	文字単位の開始位置
	文字数	取出す文字数
戻り	文字	切り出した文字列
エラー		

LEN()

文字列内の文字数を得る。

書式	変数 = LEN(文字列)	
引数	文字列	文字数を調べたい文字列
戻り	数値	文字の数（全ての文字を1文字として数える）
エラー		

30 ◎数学系関数(1/2)

◎数学系関数(1/2)

整数部の取得、乱数、絶対値や符号の取得など、数学に関する以下の関数があります。
FLOOR()、RND()、ABS()、SGN()、SQR()、EXP()、LOG()

FLOOR()

整数部を取り出す。
1バイト範囲の整数取出しには、AND 命令も利用可能です。（例）A=A AND &HFF

書式	変数 = FLOOR(数値)	
引数	数値	数値
戻り	数値	求めた結果
エラー		

RND()

指定値までの乱数を得る。

書式	変数 = RND(最大値)	
引数	最大値	生成する数値の最大値
戻り	数値	0～最大値までの乱数（最大値は含まれない）
エラー		

ABS()

絶対値を得る。

書式	変数 = ABS(数値)	
引数	数値	絶対値を取りたい数値
戻り	数値	絶対値
エラー		

SGN()

符号取得。

書式	変数 = SGN(変数)	
引数	数値	符号を確認する数値
戻り	数値	0または、±1
エラー		

30 ◎数学系関数(1/2)

SQR()

平方根値を返す。

書式	変数 = SQR(数値)	
引数	数値	平方根を求めたい数値
戻り	数値	求めた結果
エラー		

EXP()

指数値を求める。

書式	変数 = EXP(数値)	
引数	数値	指数値を求めたい数値
戻り	数値	求めた結果
エラー		

LOG()

自然対数を求める。

書式	変数 = LOG(数値)	
引数	数値	自然対数を求めたい数値
戻り	数値	求めた結果
エラー		

31 ◎数学系関数(2/2)

◎数学系関数(2/2)

サイン値、コサイン値の取得など、数学に関する以下の関数があります。
PI()、RAD()、DEG()、SIN()、COS()、TAN()、ATAN()

PI()

パイを返す。

書式	変数 = PI()	
引数	なし	
戻り	数値	円周率の値
エラー		

RAD()

角度情報からラジアン値を得る。

書式	変数 = RAD(角度)	
引数	角度	0〜360
戻り	数値	角度から求めたラジアン数値
エラー		

DEG()

ラジアン値から角度情報を得る。

書式	変数 = DEG(ラジアン値)	
引数	ラジアン値	0〜2π
戻り	数値	ラジアン値から求めた角度
エラー		

SIN()

サイン値を返す。

書式	変数 = SIN(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

31 ◎数学系関数(2/2)

COS()

コサイン値を返す。

書式	変数 = COS(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

TAN()

タンジェント値を返す。

書式	変数 = TAN(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

ATAN()

アークタンジェント値を返す。
引数を2つ（ Y, X ）渡すと移動量から方向を求める関数としても利用可能です。
求める方向=ATAN(目的y-y, 目的x-x)

書式	変数 = ATAN(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

32 エラー番号表

BASICを実行中に発生するエラーや警告表現には以下の要素があります。エラー発生時、システム変数ERRにエラー番号、ERLに行番号が記録されます。

1	Syntax error	おかしな文法が含まれています。
2	Out of range	数値が有効範囲を超えました。
3	Out of memory	メモリーが不足しています。
4	Undefined label	分岐先が見つかりません。
5	NEXT without FOR	FORに対応しないNEXTがあります。
6	Out of DATA	READで取得するためのDATAが不足しています。
7	Illegal function call	関数や命令の機能の指定方法が間違っています。
8	Duplicate definition	配列や変数を二重に定義しました。
9	Can't continue	CONTでプログラムを再開できません。
10	Missing operand	パラメータが不足しています。
11	Duplicate label	ラベルを二重に定義しました。
12	Illegal resource type	指定されたリソース種類文字列は存在しません。
13	Illegal character type	指定された種類のキャラは存在しません。
14	String too long	文字列が長すぎます。ラベルは8文字、文字列は256文字以内で収めてください。
15	Division by zero	0による除算を行いました。
16	Overflow	演算結果が許容範囲を超えました。
17	Subscript out of range	配列変数の添字が範囲外です。
18	Type mismatch	変数の型が一致しません。
19	Fomula too complex	式が複雑すぎます（括弧が多すぎるなど）。
20	RETURN without GOSUB	GOSUBがないのにRETURNがあります。
21	FOR without NEXT	NEXTに対応しないFORがあります。

33 キャラクタコード表(1/2)

SPU0

SPU1

SPU2

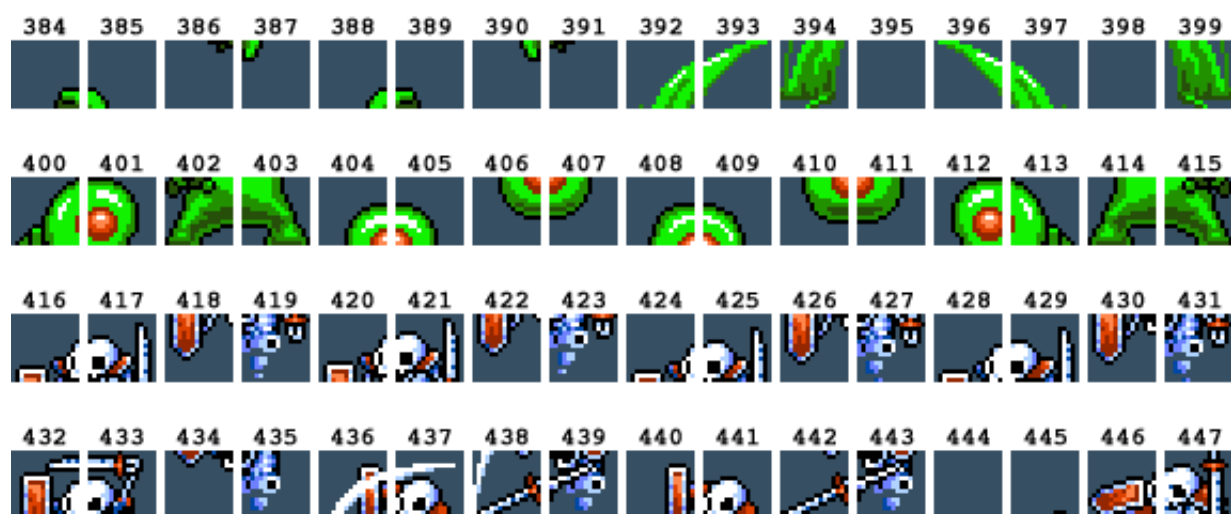
33 キャラクタコード表(1/2)

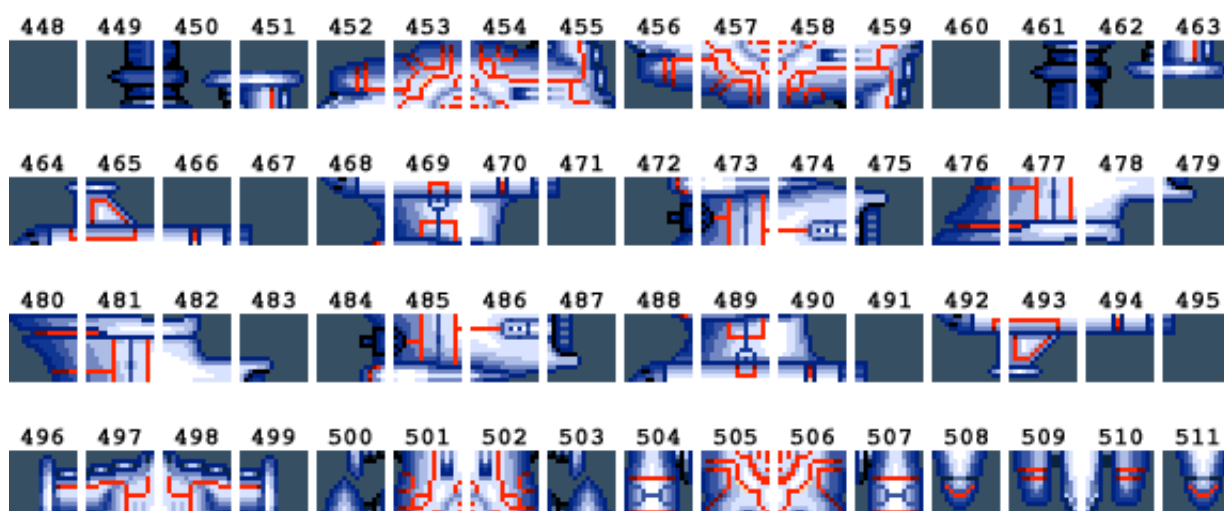
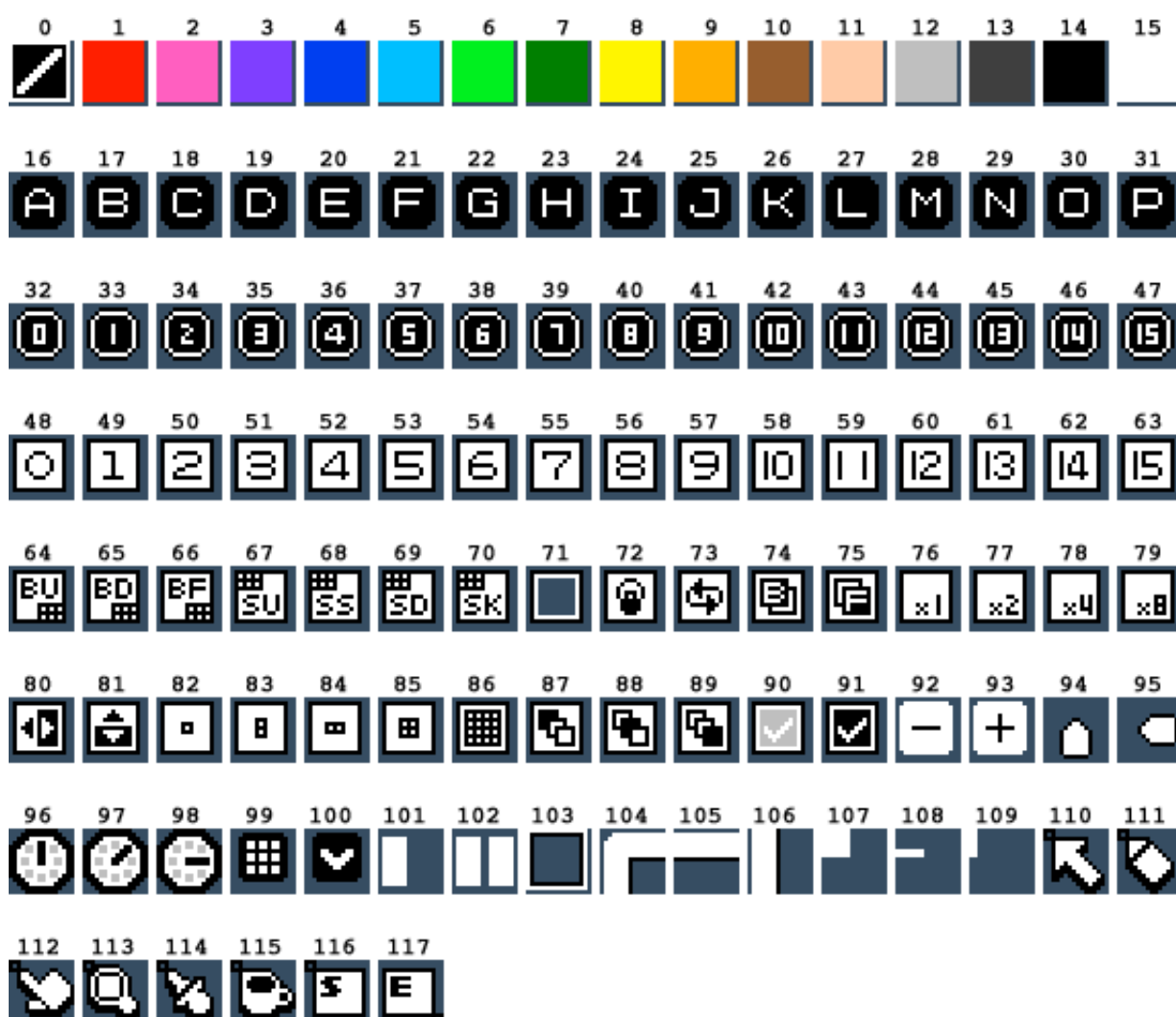
SPU3

SPU4

SPU5

33 キャラクタコード表(1/2)

SPU6

34 キャラクタコード表(2/2)

SPU7

SPS0(下画面用SPRITE)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47

48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63

64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95

96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111

112 113 114 115 116 117

34 キャラクタコード表(2/2)

BGU0

34 キャラクタコード表(2/2)

BGU2

34 キャラクタコード表(2/2)

BGU3

34 キャラクタコード表(2/2)

システムアイコン

35 用語集

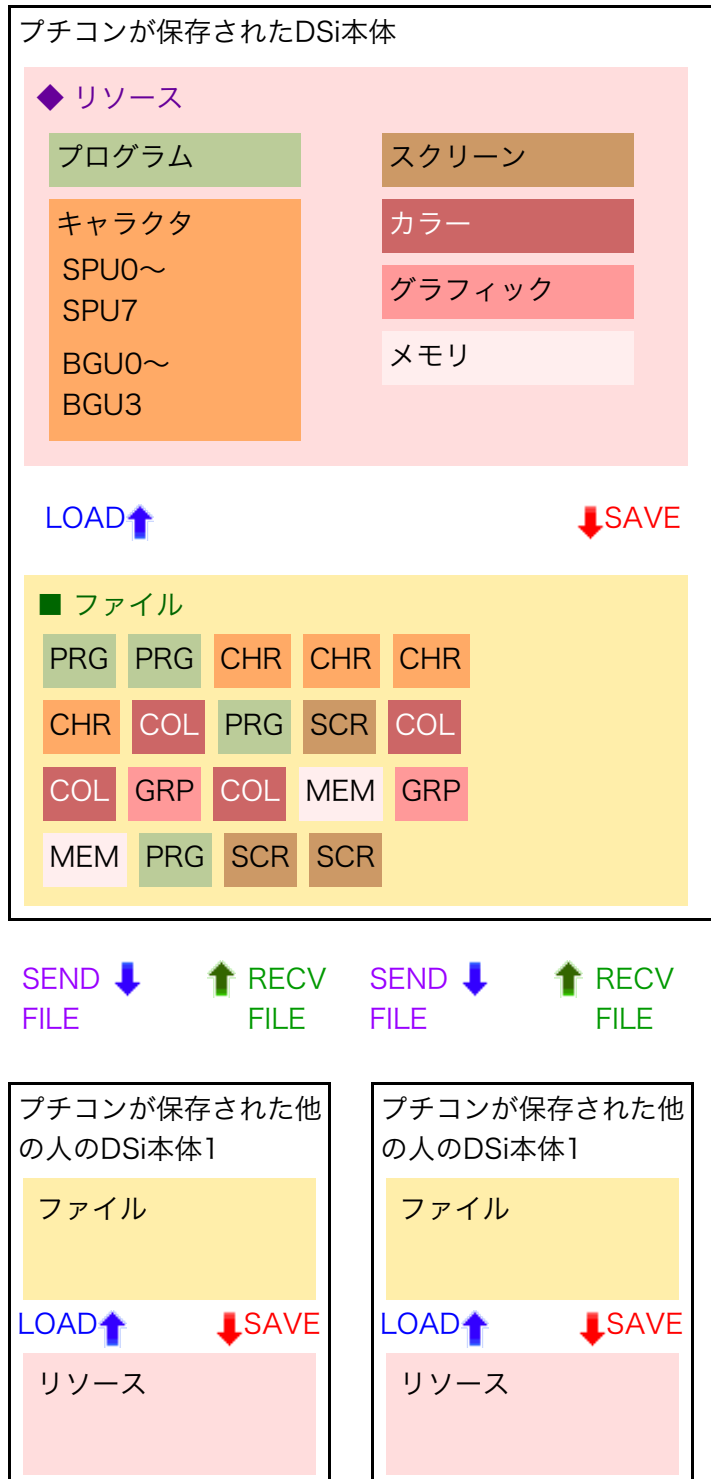
SPRITE	最大100個のアニメ付キャラを表示するための機能
カラー	BG・SPRITE・グラフィック用に256色ずつ（0番は透明）
キャラクタ	BGやSPRITEで利用する8x8ドット単位16色の画像
システムアイコン	モードやツールの切替を行うアイコンボタン
ドット	画面上の絵や文字を構成する色のついた点
パレット	画像や文字に割り当てる色単位。 1パレット＝16色（0番は透明）
フレーム	表示同期を取るための時間の単位 1フレーム＝1/60(秒)
ファイル	作成したプログラムや画像などを保存するメモリ領域
リソース	BASIC内で利用するために、ファイルから読み込んだプログラムや画像などの総称。内部的に確保されたメモリー内またはビデオメモリー内に保持される

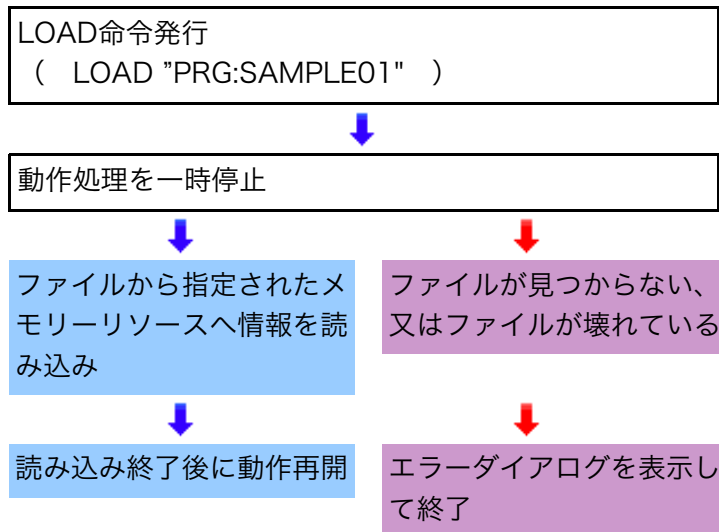
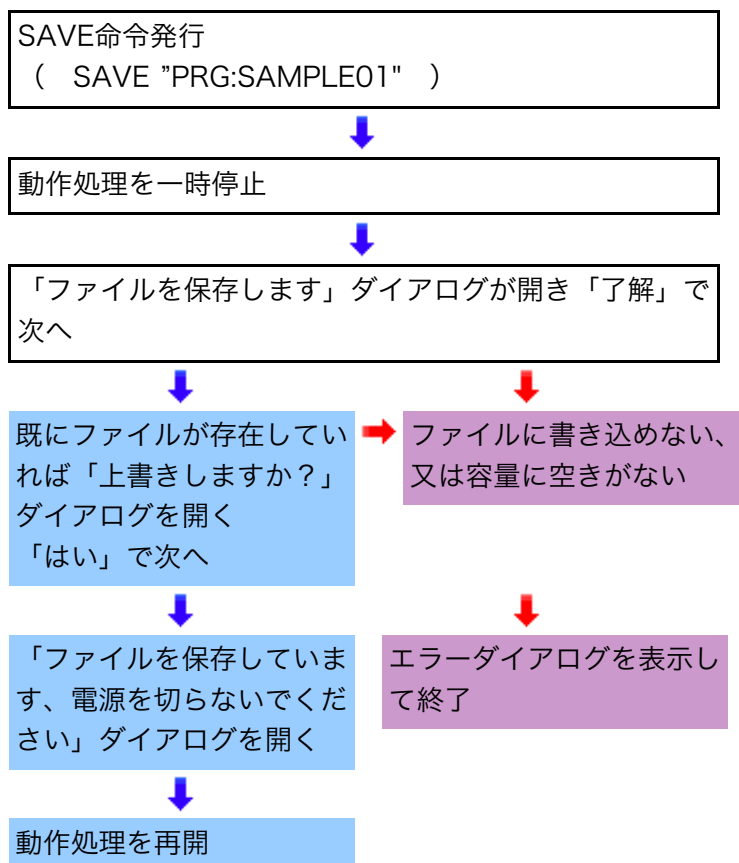
36 ファイルとリソースについて

プログラム内でファイルをアクセスする場合、およびプチコンが保存された他のDSi本体への送受信に関しての遷移を記述します。

対象となる命令
LOAD、SAVE、SENDFILE、RECVFILE

ファイルとリソースの関係

LOAD・SAVEは、内部リソースとファイル間の読み書きを行い、RECVFILE・SENDFILEは、プチコンが保存された他のDSi本体との通信によるファイルへの読み書きに使用します。

36 ファイルとリソースについて

LOADの流れ

SAVEの流れ

36 ファイルとリソースについて

SENDFILE（送信）の流れ

RECVFILE（受信）の流れ